Contents

[Problem 2](#_Toc20228662)

[Solution 3](#_Toc20228663)

[**Task 1:** Import data into Python environment. 3](#_Toc20228664)

[Task 1: Import data into Python environment. 3](#_Toc20228665)

[**Task 2:** Provide the trend chart for the number of complaints at monthly and daily granularity levels. 5](#_Toc20228666)

[Number of Complaints per Day 5](#_Toc20228667)

[Number of Complaints per Month 6](#_Toc20228668)

[**Task 3:** Provide a table with the frequency of complaint types. 8](#_Toc20228669)

[Which complaint types are maximum i.e., around internet, network issues, or across any other domains. 8](#_Toc20228670)

[**Task 4:** Create a new categorical variable with value as **Open** and **Closed**. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed. 12](#_Toc20228671)

[**Task 5:** Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on: Which state has the maximum complaints and Which state has the highest percentage of unresolved complaints 12](#_Toc20228672)

Problem

DESCRIPTION

Comcast is an American global telecommunication company. The firm has been providing terrible customer service. They continue to fall short despite repeated promises to improve. Only last month (October 2016) the authority fined them a $2.3 million, after receiving over 1000 consumer complaints.  
The existing database will serve as a repository of public customer complaints filed against Comcast.  
It will help to pin down what is wrong with Comcast's customer service.

**Data Dictionary**

* Ticket #: Ticket number assigned to each complaint
* Customer Complaint: Description of complaint
* Date: Date of complaint
* Time: Time of complaint
* Received Via: Mode of communication of the complaint
* City: Customer city
* State: Customer state
* Zipcode: Customer zip
* Status: Status of complaint
* Filing on behalf of someone

**Analysis Task**

To perform these tasks, you can use any of the different Python libraries such as NumPy, SciPy, Pandas, scikit-learn, matplotlib, and BeautifulSoup.

- Import data into Python environment.  
- Provide the trend chart for the number of complaints at monthly and daily granularity levels.  
- Provide a table with the frequency of complaint types.

* Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

- Create a new categorical variable with value as **Open** and **Closed**. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.  
- Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:

* Which state has the maximum complaints
* Which state has the highest percentage of unresolved complaints

- Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

The analysis results to be provided with insights wherever applicable.

# Solution

## **Task 1:** Import data into Python environment.

Used Pandas read\_csv command to read the dataset into python environment. Merged data for state ‘District Of Columbia’ and ‘District of Columbia’

#import necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib import style

%matplotlib inline

## Task 1: Import data into Python environment.

In [515]:

df\_comcast\_consumer\_complaints = pd.read\_csv("../Dataset/Comcast\_telecom\_complaints\_data.csv")

In [516]:

#view top 5 records

df\_comcast\_consumer\_complaints.head()

Out[516]:

|  | **Ticket #** | **Customer Complaint** | **Date** | **Date\_month\_year** | **Time** | **Received Via** | **City** | **State** | **Zip code** | **Status** | **Filing on Behalf of Someone** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland | 21009 | Closed | No |
| **1** | 223441 | Payment disappear - service got disconnected | 04-08-15 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia | 30102 | Closed | No |
| **2** | 242732 | Speed and Service | 18-04-15 | 18-Apr-15 | 9:55:47 AM | Internet | Acworth | Georgia | 30101 | Closed | Yes |
| **3** | 277946 | Comcast Imposed a New Usage Cap of 300GB that ... | 05-07-15 | 05-Jul-15 | 11:59:35 AM | Internet | Acworth | Georgia | 30101 | Open | Yes |
| **4** | 307175 | Comcast not working and no service to boot | 26-05-15 | 26-May-15 | 1:25:26 PM | Internet | Acworth | Georgia | 30101 | Solved | No |

In [361]:

df\_comcast\_consumer\_complaints.describe()

Out[361]:

|  | **Zip code** |
| --- | --- |
| **count** | 2224.000000 |
| **mean** | 47994.393435 |
| **std** | 28885.279427 |
| **min** | 1075.000000 |
| **25%** | 30056.500000 |
| **50%** | 37211.000000 |
| **75%** | 77058.750000 |
| **max** | 99223.000000 |

In [362]:

#check the dataframe size

df\_comcast\_consumer\_complaints.size

Out[362]:

24464

In [695]:

#Clean data: merge District Of Columbia and District of Columbia data

df\_comcast\_consumer\_complaints['State'] =df\_comcast\_consumer\_complaints['State'].apply(lambda x: 'District of Columbia' if x=='District Of Columbia' else x)

#check data

df\_comcast\_consumer\_complaints[df\_comcast\_consumer\_complaints['State'] =='District of Columbia'].describe()

Out[695]:

|  | **Zip code** |
| --- | --- |
| **count** | 17.000000 |
| **mean** | 20016.588235 |
| **std** | 13.915184 |
| **min** | 20001.000000 |
| **25%** | 20003.000000 |
| **50%** | 20013.000000 |
| **75%** | 20026.000000 |
| **max** | 20037.000000 |

## **Task 2:** Provide the trend chart for the number of complaints at monthly and daily granularity levels.

Used matplotlib plot to create a trend chart of complaints per day and per month

### Number of Complaints per Day

In [696]:

#get daily complaint counts

df\_daily\_complaints = df\_comcast\_consumer\_complaints.groupby(['Date'],as\_index=False).agg(['mean','count']).reset\_index()

df\_daily\_complaints.head()

Out[696]:

|  | **Date** | **Zip code** | |
| --- | --- | --- | --- |
|  |  | **mean** | **count** |
| **0** | 04-01-15 | 41808.111111 | 18 |
| **1** | 04-02-15 | 51878.703704 | 27 |
| **2** | 04-03-15 | 50994.733333 | 15 |
| **3** | 04-04-15 | 56334.833333 | 12 |
| **4** | 04-05-15 | 67932.666667 | 6 |

In [697]:

x\_axis\_data = df\_daily\_complaints.iloc[:, 0]

y\_axis\_data =df\_daily\_complaints.iloc[:, 2]

style.use('ggplot')

plt.figure(figsize=(20,4))

#plt.axis([6.5,17.5,50,2000])

plt.xticks(rotation=45, ha="right")

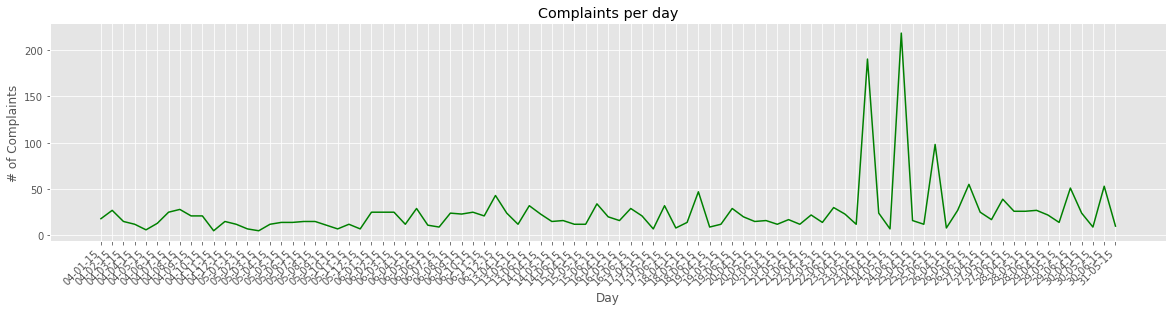
plt.plot(x\_axis\_data,y\_axis\_data,'g',linewidth=1.5)

plt.title('Complaints per day')

plt.xlabel('Day')

plt.ylabel('# of Complaints')

plt.show()



### Number of Complaints per Month

In [698]:

#define a function to get month from a date

import datetime

def getMonth(date):

return datetime.datetime.strptime(date, "%d-%m-%y").month

In [699]:

#test the function

getMonth('31-03-15')

Out[699]:

3

In [700]:

df\_comcast\_consumer\_complaints['Month\_of\_date']=df\_comcast\_consumer\_complaints['Date'].apply(getMonth)

df\_comcast\_consumer\_complaints.head(2)

Out[700]:

|  | **Ticket #** | **Customer Complaint** | **Date** | **Date\_month\_year** | **Time** | **Received Via** | **City** | **State** | **Zip code** | **Status** | **Filing on Behalf of Someone** | **Ticket\_Status** | **Complaint\_Type** | **Month\_of\_date** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland | 21009 | Closed | No | Closed | Internet | 4 |
| **1** | 223441 | Payment disappear - service got disconnected | 04-08-15 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia | 30102 | Closed | No | Closed | Other | 8 |

In [701]:

#get monthly complaint counts

df\_monthly\_complaints = df\_comcast\_consumer\_complaints.groupby(["Month\_of\_date"],as\_index=False).agg(['mean', 'count']).reset\_index()

df\_monthly\_complaints.head(2)

Out[701]:

|  | **Month\_of\_date** | **Zip code** | |
| --- | --- | --- | --- |
|  |  | **mean** | **count** |
| **0** | 1 | 45154.890909 | 55 |
| **1** | 2 | 51229.661017 | 59 |

In [702]:

*#plot graph*

**import** **calendar**

x\_axis\_data = df\_monthly\_complaints.iloc[:, 0].apply(**lambda** x: calendar.month\_abbr[x])

y\_axis\_data =df\_monthly\_complaints.iloc[:, 2]

style.use("ggplot")

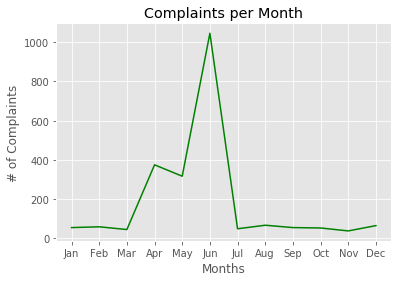
plt.plot(x\_axis\_data,y\_axis\_data,'g',linewidth=1.5)

plt.xlabel('Months')

plt.ylabel('# of Complaints')

plt.title('Complaints per Month')

plt.show()



## **Task 3:** Provide a table with the frequency of complaint types.

## Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

Used string and nltk.corpus stopwords to remove punctuation and stopwords and separate out words in column ‘Customer Complaints’. Added the logic in a function.

Added and used a function to check for keywords to distinguish between type of Complaints.

Used Group by and count to get the frequency table and maximum Complaint Type.

df\_comcast\_consumer\_complaints.head()

Out[722]:

|  | **Ticket #** | **Customer Complaint** | **Date** | **Date\_month\_year** | **Time** | **Received Via** | **City** | **State** | **Zip code** | **Status** | **Filing on Behalf of Someone** | **Ticket\_Status** | **Complaint\_Type** | **Month\_of\_date** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland | 21009 | Closed | No | Closed | Internet | 4 |
| **1** | 223441 | Payment disappear - service got disconnected | 04-08-15 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia | 30102 | Closed | No | Closed | Other | 8 |
| **2** | 242732 | Speed and Service | 18-04-15 | 18-Apr-15 | 9:55:47 AM | Internet | Acworth | Georgia | 30101 | Closed | Yes | Closed | Network | 4 |
| **3** | 277946 | Comcast Imposed a New Usage Cap of 300GB that ... | 05-07-15 | 05-Jul-15 | 11:59:35 AM | Internet | Acworth | Georgia | 30101 | Open | Yes | Open | Other | 7 |
| **4** | 307175 | Comcast not working and no service to boot | 26-05-15 | 26-May-15 | 1:25:26 PM | Internet | Acworth | Georgia | 30101 | Solved | No | Closed | Other | 5 |

In [704]:

#Import necessary Libraries

import string

from nltk.corpus import stopwords

In [705]:

#define a function to get rid of stopwords present in the messages

def message\_text\_process(mess):

no\_punctuation=[char for char in mess if char not in string.punctuation]

no\_punctuation =''.join(no\_punctuation)

return ["""'""" + word +"""'""" for word in no\_punctuation.split() if word not in stopwords.words('english') ]

In [706]:

#verify that function is working

df\_complaints\_types= df\_comcast\_consumer\_complaints

df\_complaints\_types['Customer Complaint'].head(5).apply(message\_text\_process)

Out[706]:

0 ['Comcast', 'Cable', 'Internet', 'Speeds']

1 ['Payment', 'disappear', 'service', 'got', 'di...

2 ['Speed', 'Service']

3 ['Comcast', 'Imposed', 'New', 'Usage', 'Cap', ...

4 ['Comcast', 'working', 'service', 'boot']

Name: Customer Complaint, dtype: object

In [707]:

#apply the function on the dataframe

df['Customer Complaint']=df\_complaints\_types['Customer Complaint'].apply(message\_text\_process)

In [708]:

df.head()

Out[708]:

Complaint\_Type

Internet 532

Network 233

Other 1459

Customer Complaint 0 ['Comcast', 'Cable', 'Internet'...

Name: Ticket #, dtype: object

In [709]:

def TypeOfIssue(mylist):

# mylist = mylist.apply(lambda s:s.lower() if type(s) == str else s)

mylist=mylist.lower()

if mylist.count('internet') >0:

return 'Internet'

elif mylist.count('network')>0 or mylist.count('bandwidth')>0 or mylist.count('data cap')>0 or mylist.count('speed')>0:

return 'Network'

else:

return 'Other'

In [710]:

#verify Function

IsInternetIssue(['Comcast', 'Cable', 'Internet', 'Speeds'])

Out[710]:

'Internet'

In [723]:

df\_complaints\_types['Complaint\_Type'] = df\_complaints\_types['Customer Complaint'].apply(TypeOfIssue)

df\_complaints\_types.head()

Out[723]:

|  | **Ticket #** | **Customer Complaint** | **Date** | **Date\_month\_year** | **Time** | **Received Via** | **City** | **State** | **Zip code** | **Status** | **Filing on Behalf of Someone** | **Ticket\_Status** | **Complaint\_Type** | **Month\_of\_date** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland | 21009 | Closed | No | Closed | Internet | 4 |
| **1** | 223441 | Payment disappear - service got disconnected | 04-08-15 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia | 30102 | Closed | No | Closed | Other | 8 |
| **2** | 242732 | Speed and Service | 18-04-15 | 18-Apr-15 | 9:55:47 AM | Internet | Acworth | Georgia | 30101 | Closed | Yes | Closed | Network | 4 |
| **3** | 277946 | Comcast Imposed a New Usage Cap of 300GB that ... | 05-07-15 | 05-Jul-15 | 11:59:35 AM | Internet | Acworth | Georgia | 30101 | Open | Yes | Open | Other | 7 |
| **4** | 307175 | Comcast not working and no service to boot | 26-05-15 | 26-May-15 | 1:25:26 PM | Internet | Acworth | Georgia | 30101 | Solved | No | Closed | Other | 5 |

In [712]:

#df\_complaints\_types.groupby(['Complaint\_Type'],as\_index=False).agg({"Total":'size'})

df = df\_complaints\_types.groupby('Complaint\_Type')['Ticket #'].count()

df

Out[712]:

Complaint\_Type

Internet 532

Network 233

Other 1459

Name: Ticket #, dtype: int64

In [713]:

# Data Insight:

# Which complaint types are maximum? : Other

df.idxmax()

Out[713]:

'Other'

## **Task 4:** Create a new categorical variable with value as **Open** and **Closed**. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.

Added a function to check for values in ‘Status’ column and categorize the Status between ‘Open’ and ‘Closed’. Applied the function to the dataframe and added a new column to the DataFrame ‘Ticket\_Status’

*#define a function*

**def** openclose(val):

**if**(val.lower()=='open' **or** val.lower()=='pending'):

**return** 'Open'

**else**:

**return** 'Closed'

In [715]:

df\_comcast\_consumer\_complaints['Ticket\_Status']=df\_comcast\_consumer\_complaints['Status'].apply(openclose)

## **Task 5:** Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on: Which state has the maximum complaints and Which state has the highest percentage of unresolved complaints

Grouped by ‘State’ and got Total, Closed and Open ticket counts and open ticket percentage.

Used matplotlib bar to create a stacked bar chart.

df = (df\_comcast\_consumer\_complaints.assign(

A=(df\_comcast\_consumer\_complaints['Ticket\_Status']== 'Open')

)

.groupby(['State'])['A']

.agg([('Open','sum'),('Total','size')])

.astype(int)

.reset\_index())

df= df.assign(Closed=(df['Total']-df['Open']))

#print(df.iloc[:,0])

df=df.assign(PercentOpen = ((df['Open']/df['Total'])\*100))

df.head(5)

Out[716]:

|  | **State** | **Open** | **Total** | **Closed** | **PercentOpen** |
| --- | --- | --- | --- | --- | --- |
| **0** | Alabama | 9 | 26 | 17 | 34.615385 |
| **1** | Arizona | 6 | 20 | 14 | 30.000000 |
| **2** | Arkansas | 0 | 6 | 6 | 0.000000 |
| **3** | California | 61 | 220 | 159 | 27.727273 |
| **4** | Colorado | 22 | 80 | 58 | 27.500000 |

In [717]:

#Plot Graph

states=np.array(df.iloc[:,0])

closed\_complaints = np.array(df.iloc[:,3])

Open\_complaints = np.array(df.iloc[:,1])

style.use('ggplot')

plt.figure(figsize=(20,4))

plt.bar(states, closed\_complaints, width=0.6, label='Closed', color='gold')

plt.bar(states, Open\_complaints, width=0.6, label='Open', color='silver')

plt.xticks(states, states)

plt.xticks(rotation=45, ha="right")

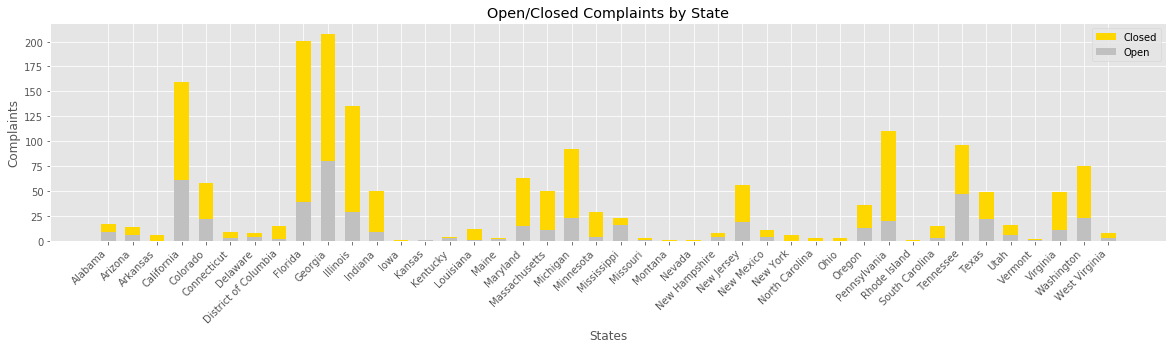
plt.ylabel("Complaints")

plt.xlabel("States")

plt.legend(loc="upper right")

plt.title("Open/Closed Complaints by State")

plt.show()



In [718]:

#Data Insight:

#The state with maximun complaints : Georgia

print("The state with maximun complaints:")

print(df.loc[df['Total'].idxmax()])

#The state with highest percentage of unresolved Complaints: Kansas

print("\nThe state with highest percentage of unresolved Complaints:")

print(df.loc[df['PercentOpen'].idxmax()])

The state with maximun complaints:

State Georgia

Open 80

Total 288

Closed 208

PercentOpen 27.7778

Name: 9, dtype: object

The state with highest percentage of unresolved Complaints:

State Kansas

Open 1

Total 2

Closed 1

PercentOpen 50

Name: 13, dtype: object

**Task 6:** Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.

Grouped by ‘Received Via’ and got Total, Closed, PercentageClosed and PercentageOpen.

df\_comcast\_consumer\_complaints.head(2)

Out[719]:

|  | **Ticket #** | **Customer Complaint** | **Date** | **Date\_month\_year** | **Time** | **Received Via** | **City** | **State** | **Zip code** | **Status** | **Filing on Behalf of Someone** | **Ticket\_Status** | **Complaint\_Type** | **Month\_of\_date** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 250635 | Comcast Cable Internet Speeds | 22-04-15 | 22-Apr-15 | 3:53:50 PM | Customer Care Call | Abingdon | Maryland | 21009 | Closed | No | Closed | Internet | 4 |
| **1** | 223441 | Payment disappear - service got disconnected | 04-08-15 | 04-Aug-15 | 10:22:56 AM | Internet | Acworth | Georgia | 30102 | Closed | No | Closed | Other | 8 |

In [720]:

#Create a new dataframe of Received Via and Ticket\_Status columns

df\_complaints\_ReceivedBy=df\_comcast\_consumer\_complaints[['Received Via','Ticket\_Status']]

df\_complaints\_ReceivedBy.head(2)

Out[720]:

|  | **Received Via** | **Ticket\_Status** |
| --- | --- | --- |
| **0** | Customer Care Call | Closed |
| **1** | Internet | Closed |

In [721]:

#df\_complaints\_ReceivedBy.groupby(['Received Via'],as\_index=False).describe()

df = (df\_complaints\_ReceivedBy.assign(

A=(df\_complaints\_ReceivedBy['Ticket\_Status']== 'Open')

)

.groupby(['Received Via'])['A']

.agg([('Open','sum'),('Total','size')])

.astype(int)

.reset\_index())

df= df.assign(Closed=(df['Total']-df['Open']))

#print(df.iloc[:,0])

df=df.assign(PercentResolved = ((df['Open']/df['Total'])\*100))

df=df.assign(PercentClosed = ((df['Closed']/df['Total'])\*100))

df

Out[721]:

|  | **Received Via** | **Open** | **Total** | **Closed** | **PercentResolved** | **PercentClosed** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | Customer Care Call | 255 | 1119 | 864 | 22.788204 | 77.211796 |
| **1** | Internet | 262 | 1105 | 843 | 23.710407 | 76.289593 |

In [529]:

#data Insight:

#Percentage of complaints resolved till date:

# Internet: 23.71%

# Customer Care Call:22.79%